# MRSP: Learn Multi-Representations of Single Primitive for Compositional Zero-Shot Learning

Dongyao Jiang ⋆, Hui Chen, Haodong Jing, Yongqiang Ma, and Nanning Zheng(✉)

National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Shanxi, China
jdy20020305@stu.xjtu.edu.cn, nnzheng@xjtu.edu.cn

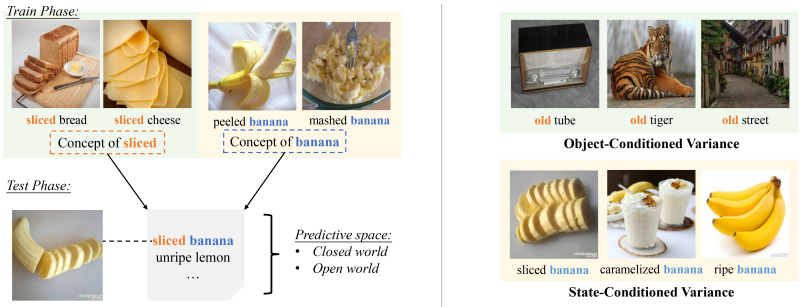
**Abstract.** Compositional Zero-Shot Learning (CZSL) aims to classify unseen state-object compositions using seen primitives. Previous methods commonly map an identical primitive from different compositions to the same area within embedding space, aiming to establish primitive representation or assess decoding proficiency. However, relying solely on the intersection area of primitive concepts might overlook nuanced semantics due to conditional variance, thereby limiting the model's capacity to generalize to unseen compositions. In contrast, our approach constructs primitive representations by considering the union area of primitives. We propose a Multiple Representation of Single Primitive learning framework (termed MRSP) for CZSL, which captures composition-relevant features through a state-object-composition three-branch cross-attention architecture. Specifically, the input image feature cross-attends to multiple state, object, and composition features and the prediction scores are adaptively adjusted by combining the output of each branch. Extensive experiments on three benchmarks in both closed-world and open-world settings showcase the superior effectiveness of MRSP.

**Keywords:** Compositional Zero-Shot Learning · Attention · Graph Convolution Networks · Vision-Language Models

## 1 Introduction

Human beings effortlessly merge familiar visual primitives to form novel composition concepts, such as *purple apple*, drawing on their comprehension of states like *purple* and objects like *apple*. However, imparting this capacity to machines to recognize zero-shot compositions, where they must generate unseen composition features by coupling state and object primitives, remains a formidable challenge. This problem is termed Compositional Zero-Shot Learning (CZSL) [4,22], which

⋆ ✉ Corresponding author. This research received funding from the National Natural Science Foundation of China through Grants 62088102 and STI2030-Major Projects No.2022ZD0208801.




**(a)** CZSL setting **(b)** conditional variance

**Fig. 1: CZSL Task**: (a) Models aim to classify compositional concepts such as *sliced banana* without prior knowledge, relying solely on provided instances like *sliced*, *banana*. (b) Conditional variance refers to the phenomenon where individual primitives display varied visual characteristics when combined.

aims to classify unseen compositions relying on seen state and object primitives, as shown in Fig. 1a.

The key to CZSL lies in learning exhaustive primitive representations. Previous methods [4, 18, 28] learn to disentangle compositions into single primitive, where each primitive is represented by single representation. Then, these primitives are coupled into compositions with a coupling module. During inference, the module is employed to generate unseen composition features for recognition. However, we argue that the approach of associating a primitive with single representation may not sufficiently enable the model to generalize to unseen compositions. This is because a primitive may only manifest a portion of its significance within a composition. For example, while both *old tiger* and *old town* share *old* primitive, the former denotes *senior*, whereas the latter suggests *historical*. This phenomenon, as shown in Fig. 1b, is known as conditional variance [8].

If excluding conditional variance, the learning process of a primitive is akin to identify a compact region in the semantic space that holds a dense cluster of training samples, thus acting as an **intersection** area for semantic information. However, considering conditional variance, a primitive can exhibit multiple distinct semantics concurrently, enclosing all these in a larger region introduces incorrect information. Instead, leveraging the **union** of several disjoint regions markedly enhances the model's ability to accurately capture the concept.

In this work, we propose a **M**ultiple **R**epresentation of **S**ingle **P**rimitive framework (MRSP) for CZSL that is designed to learn union embeddings for each primitive. To this end, MRSP utilizes graph convolution networks (GCN) to encode primitive and composition features and cross-attention decoding to output prediction scores. Specifically, each node of the GCN module represents a primitive or composition concept, and the feature of each node is extracted with the CLIP text encoder. In the compositional graph, a primitive is represented by multiple nodes, and each node is initialized with the primitive representation and exhibits distinct features after graph convolutional operations.

After learning multiple representations of the primitives, we utilize three cross-attention [3] branches: the composition branch, state branch and object branch. The composition branch receives inputs of composition representation and image features, while the state (object) branch receives inputs of state (object) representations and image features. We employ cross-attention to model semantic correlations between the two modal inputs, with each branch learning to assign higher weights to the correct representations. By supervising the representational capacity of the branch outputs, the model ultimately comprehends the interaction of primitives within seen compositions and achieves remarkable performance on unseen compositions by leveraging multiple representations.

Mancini *et al*. [20] introduced a more challenging open-world setup that expands the prediction space to encompass the Cartesian product of the state and object set. To assess the effectiveness of MRSP, we conduct extensive experiments on the MIT-States, UT-Zappos, and Clothing16K datasets, under the closed-world and open-world settings. The results demonstrate that MRSP achieves state-of-art performance across all these benchmarks. The contributions of this work can be summarized as follows:

- We introduce a novel framework named MRSP which leverages **M**ultiple **R**epresentation to construct the union of **S**ingle **P**rimitive concepts.
- The multiple representation construction module learns from the **union** set of language features fused with images features to enhance the model's ability of generalizing to unseen compositions. We also design a three-branch cross-attention decoder for decoding both seen and unseen compositions, enabling concept decomposition at the attention level.
- Extensive results across multiple datasets indicate that MRSP achieves state-of-the-art performance, highlighting its efficiency in recognizing unseen state-object compositions.

## 2 Related Work

**Compositional Zero-Shot Learning** seeks to elucidate the interplay between states and object compositions, facilitating flexible information transfer from seen to unseen compositions. Various approaches in this domain employ diverse strategies. For instance, some methods measure distance by transforming and coupling elements, projecting them into a unified hidden space alongside visual images [4, 23, 24, 28]. Pioneering work by Misra *et al*. [22] showcased the feasibility of computers learning compositional concepts through decoupling. Karthik *et al*. [11] leverage external knowledge bases, akin to large language models, employing designs such as `Can a xxx be xxx?` to tackle a range of compositional challenges in open-world scenarios. Additionally, Hao *et al*. [8] utilize a cross-attention mechanism to establish correlations between positive and negative sample instances, serving as the foundation for subsequent model discrimination. Lastly, Li *et al*. [14] have recently pioneered the use of large language model-generated cue words to aid CZSL model understanding. As shown in

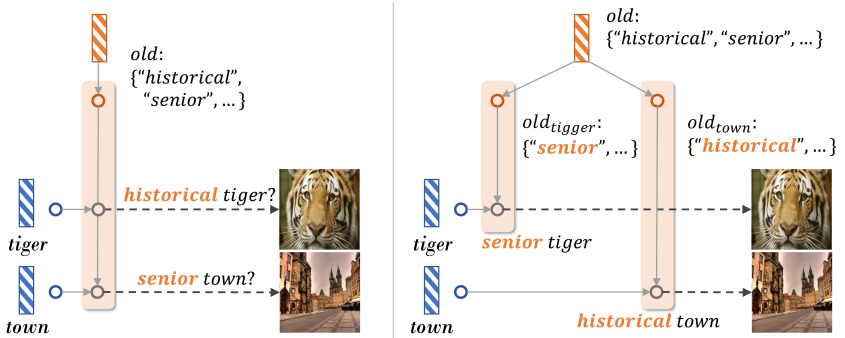



**Fig. 2:** Comparison between intersection-based methods and our union-based method, MRSP: (a) Single representation result in inappropriate compositional information. (b) Multiple representations yield more accurate composition concepts.

Fig. 2, our work differs significantly from previous studies in that our focus is on achieving a comprehensive representation of primitives, rather than solely aiming to extract the maximal concept intersection.

**Graph Convolutional Networks** were proposed by Kipf *et al.* [13], leverage hierarchical structures embedded in data to generate features, enabling the exploitation of feature hierarchies from supervised learning data. However, GCN methods often encounter limitations due to the phenomenon of `over-smoothing`, hindering their scalability. Criticisms of GCN methods have been voiced by Felix et al. [30] and Zhewei *et al.* [5]. Inspired by techniques such as residual structures, dropout layers, and skip connections, recent advancements in GCNs [5,30] have significantly broadened the scope of applications for graph-based methods.

**Attention mechanisms**, as demonstrated by Transformer [29], are powerful tools for extracting sequence relations and have achieved remarkable success in natural language processing. Building upon this, Alexey *et al.* [6] applied transformer architectures to computer vision tasks, showcasing the attention mechanism's ability to extract global image features that rival or even surpass those extracted by convolutional methods. Inspired by these advancements [3,8], we leverage cross-attention as a model for relationship extraction in images.

**Vision-Language Models** pretrained on large datasets can effectively learn semantic coherence between image and text modalities across different levels [27]. This presents an appealing avenue for leveraging textual modality data to enhance open-world visual comprehension. This work builds upon the pre-trained CLIP [7] image-text encoder.

## 3 Method

In Sec. 3.1, we formalize the CZSL problem and delineate open-world and closed-world settings. Sec. 3.2 elaborates on the construction of multiple representations

from text using GCN, aimed at deriving the union of primitive concepts as well as seen and unseen composition concepts. Following that, in Sec. 3.3, we delve into the utilization of cross-attention for decoding primitives and composition concepts from images, yielding decoding results at the attention level. Sec. 3.4 outlines the training objectives and the utilization of attention-level decoding results for prediction purposes. The entire framework is illustrated in Fig. 3.

### 3.1 Problem Formulation

Given the training set $T = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{Y}_{seen}\}$, where $x$ is an image in the RGB image space $\mathcal{X}$; $y$ is a compositional label in seen class set $\mathcal{Y}_{seen}$; composition label set $\mathcal{Y} = \mathcal{Y}_{seen} \cup \mathcal{Y}_{unseen}$. A compositional label $y$ consists of two parts $y = (s, o)$, where $s \in \mathcal{S}$ is a state primitive in the state primitive space $\mathcal{S}$;$o \in \mathcal{O}$ is an object primitive in the object primitive space $\mathcal{O}$, and all primitive elements in both spaces $\mathcal{S}$ and $\mathcal{O}$ are included in the seen compositions. CZSL attempts to classify an image instance labeled $y_{unseen} \in \mathcal{Y}_{unseen}$. The setting of the size of the model prediction space can be divided into `Closed-world` [22] setting and `Open-world` [20] setting. For `Closed-world` setting, the prediction space is $\mathcal{Y}_{pred} = \mathcal{Y}_{test} = \mathcal{Y}_{seen} \cup \mathcal{Y}_{unseen}$. For `Open-world` setting, the prediction space is $\mathcal{Y}_{pred} = \mathcal{S} \times \mathcal{O}$ with $\times$ represents the Cartesian product.

In this work, we denote the text encoder as $\phi(\cdot)$ and the image encoder as $\psi(\cdot)$. Let $p_j$ represent the feature vector of a concept $j \in \mathcal{S} \cup \mathcal{O} \cup \mathcal{Y}_{pred}$ after encoding with $\phi(\cdot)$, and let the compositions containing $p_j$ in the prediction space constitute $\mathcal{Y}_{p_j}$. Let $p_j^{y_i}$ represent a subset representation associated with the combination $y_i = (s_i, o_i)$, where $y_i \in \mathcal{Y}_{p_j}$. We use $\mathcal{P}_j$ to denote the union of $p_j$, i.e., $\mathcal{P}_j = \bigcup_{y_i \in \mathcal{Y}} p_j^{y_i}$. Let $v_x$ denotes the feature vector of input $x$ after encoding with $\psi(\cdot)$.

### 3.2 Multiple Representation Construct via GCN

Training a model to learn multiple representations for each primitive concept from scratch is computationally demanding. Alternatively, a more efficient strategy involves extracting relevant subsets from comprehensive CLIP [27] representations. Subsequently, we utilize graphs to delineate topological relationships, enabling the derivation of the union of primitive and composition concepts.

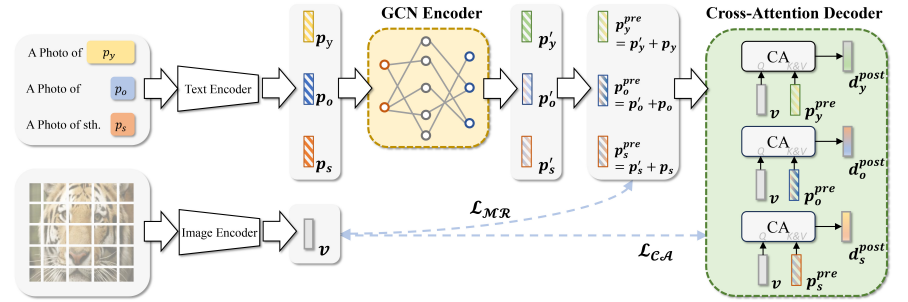
**Primitives and Compositions Embedding.** Given a label text $s \in S$, $o \in O$, or $y \in \mathcal{Y}_{pred}$, we utilize a pre-trained CLIP text encoder [7], denoted as $\phi(\cdot)$, to generate corresponding representations $p_s, p_o, p_y$ as shown in Eqs. (1) to (3):

$$p_s = \phi(\text{“}A\ photo\ of\ something\ [s].\text{”}), \tag{1}$$

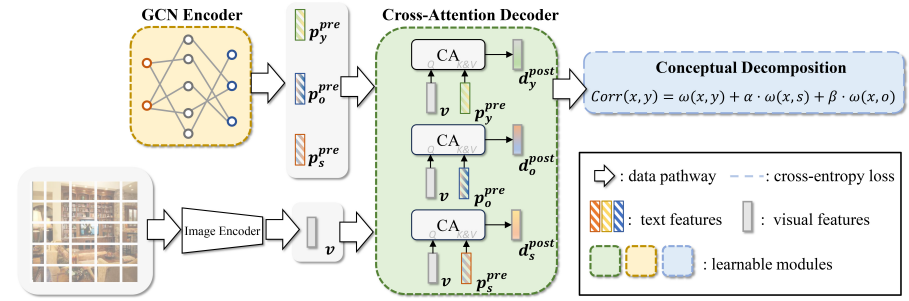
$$p_o = \phi(\text{“}A\ photo\ of\ [o].\text{”}), \tag{2}$$

$$p_y = \phi(\text{“}A\ photo\ of\ [s]\ [o].\text{”}). \tag{3}$$

Prompt design is not the focus of this work, therefore, we use the simplest hard prompt to generate $p_j$.




**(a)** Train of MRSP.




**(b)** Inference of MRSP.

**Fig. 3: MRSP Architecture.** (a) During training, the model initializes a compositional graph based on the prediction space, updates node features using a GCN module, and evaluates the quality of multiple representations with image features. Subsequently, both node and image features are inputted into a three-branch cross-attention decoder, with decoding quality assessed using image features. (b) During inference, pre-constructed multiple representations and image features are fed into the three-branch cross-attention decoder, considering decoding results from state, object, and composition branches to obtain the final inference at the attention level.

**Node Embedding.** We employ copies of $p_j$ to initialize $\mathcal{P}_j$ and $p_j^{y_i}$ with $y_i \in \mathcal{Y}_{p_j}$, obtaining $\mathcal{P}_j = p_j \in \mathbb{R}^d$ and $\mathcal{P}_j^{\mathcal{Y}} = [p_j; \ldots; p_j] \in \mathbb{R}^{(|\mathcal{Y}_{p_j}| \times d)}$ where $d$ is the dimension of features encoded by $\phi(\cdot)$. Since the nodes in the composition graph include the universal representation $\mathcal{P}_j$ and the multiple representations $\mathcal{P}_j^{y_i}$, we can initialize all nodes as shown in Eqs. (4) to (7):

$$H_s = [\mathcal{P}_{s_1}; \mathcal{P}_{s_1}^{\mathcal{Y}}; \ldots; \mathcal{P}_{s_{|\mathcal{S}|}}; \mathcal{P}_{s_{|\mathcal{S}|}}^{\mathcal{Y}}] \in \mathbb{R}^{(|\mathcal{S}|+N_{\mathcal{S}}) \times d}, \tag{4}$$

$$H_o = [\mathcal{P}_{o_1}; \mathcal{P}_{o_1}^{\mathcal{Y}}; \ldots; \mathcal{P}_{o_{|\mathcal{O}|}}; \mathcal{P}_{o_{|\mathcal{O}|}}^{\mathcal{Y}}] \in \mathbb{R}^{(|\mathcal{O}|+N_{\mathcal{O}}) \times d}, \tag{5}$$

$$H_y = [\mathcal{P}_{y_1}; \ldots; \mathcal{P}_{y_{|\mathcal{Y}|}}] \in \mathbb{R}^{|\mathcal{Y}| \times d}, \tag{6}$$

$$H^{(0)} = [H_s; H_o; H_y] \in \mathbb{R}^{(|\mathcal{S}|+|\mathcal{O}|+|\mathcal{Y}|+N_{\mathcal{O}}+N_{\mathcal{S}}) \times d}, \tag{7}$$

where $N_{\mathcal{S}} = \sum_{i=1}^{|\mathcal{S}|} |\mathcal{Y}_{s_i}|$ and $N_{\mathcal{O}} = \sum_{j=1}^{|\mathcal{O}|} |\mathcal{Y}_{o_j}|$. Clearly, $N_{\mathcal{S}} + N_{\mathcal{O}} = 2 \cdot |\mathcal{Y}|$, so we ultimately obtain $H^{(0)} \in \mathbb{R}^{(|\mathcal{S}|+|\mathcal{O}|+3 \cdot |\mathcal{Y}|) \times d}$ as the initial node features.

**Compositional Graph Constructions.** We aim to internalize compositional dependencies within the graph to assist the model in leveraging pertinent information. The nodes in the graph consist of representations for each primitive and each composition, resulting in a total of $K = |\mathcal{S}|+N_{\mathcal{S}}+|\mathcal{O}|+N_{\mathcal{O}}+|\mathcal{Y}| = |\mathcal{S}|+|\mathcal{O}|+3\cdot|\mathcal{Y}|$ nodes. As shown in Fig. 4, the construction of compositional graph can be described as: given a $y_i = (s_j, o_k) \in \mathcal{Y}_{pred}$, it effects 5 nodes include $s_j$, $y_i$, $o_k$, and $s_{j_{o_i}}$, $o_{k_{s_i}}$, connecting them with undirected edges $(s_j \leftrightarrow s_{j_{o_i}})$,$(s_{j_{o_i}} \leftrightarrow y_i)$, $(y_i \leftrightarrow o_{k_{s_i}})$, and $(o_{k_{s_i}} \leftrightarrow o_k)$. Ultimately, this yields an adjacency matrix $\widetilde{A} \in \mathbb{R}^{K\times K}$ and degree matrix $\widetilde{D} \in \mathbb{R}^{K\times K}$.

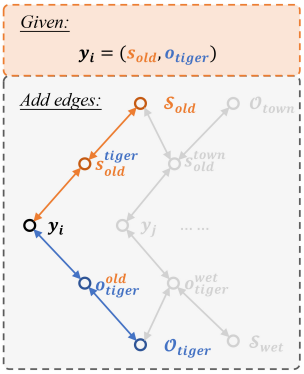



**Fig. 4:** Constructions of graph. Colored edges are added edges, colored nodes are affected nodes, gray parts are unaffected graph.

**GCN Module.** With the adjacency matrix, degree matrix and node features, we can update the concepts using Eq. (8):

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)};\Theta^{(l)}) \in \mathbb{R}^{K\times d}, \tag{8}$$

where $H^{(l)}$ denotes the node features of the $l^{th}$ layer, $\sigma$ represents the ReLU activation function, $\widetilde{A}$ is the adjacency matrix, $\widetilde{D}$ is the node degree matrix with row sums of $\widetilde{A}$ as its diagonal elements, and $\Theta^{(l)}$ indicates the learnable parameters of the $l^{th}$ layer.

Similar to Eqs. (4) to (7), we can obtain updated concept representations $p'_s$, $p'_o$, and $p'_y$ from $H^{(l)}$. After that, we utilize residual connections to derive the final concept representations, as shown in Eq. (9):

$$p^{pre}_s = p'_s + p_s, \quad p^{pre}_o = p'_o + p_o, \quad p^{pre}_y = p'_y + p_y. \tag{9}$$

To assess the quality of representations, we construct a cross-entropy loss based on this, as shown in Eqs. (10) and (11):

$$u = H^{(N)}v_x \in \mathbb{R}^K, \quad \hat{q}_i = \frac{\exp(u_i/\tau)}{\sum_{j=1}^{K}\exp(u_j/\tau)} \in \mathbb{R}, \tag{10}$$

$$\mathcal{L}_{\mathcal{MR}} = -\sum_{i=1}^{K}\mathbb{I}(i \in y)\log\hat{q}_i, \tag{11}$$

where $H^{(N)}$ is the node feature matrix outputted by the last layer of GCN, $v_x$ calculated by Eq. (12), $\tau$ is the temperature coefficient, $\mathbb{I}(p_n \in y)$ equals 1 if the condition inside the parentheses is met, otherwise 0.

### 3.3 Multiple Representation Decoding via Cross-Attention

We utilize a three-branch cross-attention [3] decoder to decode composition concepts from images, enabling concept decomposition at the attention level by supervising the representation capabilities of each branch output.

**Visual Embedding.** Given an image input $x$, we employ a pre-trained CLIP [7] image encoder followed by a two-layer MLP to generate visual features $v_x$:

$$v_x = MLP(\psi(x); \Omega_v) \in \mathbb{R}^d, \tag{12}$$

where $\Omega_v$ denotes the parameters of MLP.

**Cross-Attention Module.** For the decoding of states, objects, and compositions, we employ three distinct multi-head cross-attention branches. Each branch shares a similar structure but differs in inputs and parameters. The state branch takes $p_s^{pre}$ and $v_x$ as inputs, the object branch receives $p_o^{pre}$ and $v_x$ as inputs, and the composition branch takes $p_y^{pre}$ and $v_x$ as inputs. In each branch, query tokens are generated from the image features $v_x$, while key-value tokens are derived from the text features $p^{pre}$, as shown in Eq. (13):

$$q_v = v_x W_i^q, \quad k_t = p^{pre} W_i^k, \quad v_t = p^{pre} W_i^v, \quad i = 1, 2, \ldots, h, \tag{13}$$

where $h$ represents the number of attention heads, and $W_i^q$, $W_i^k$, and $W_i^v \in \mathbb{R}^{d \times (d/h)}$ are weight matrices. The outputs from each attention head $d_i^{post}$ are concatenated to form $d^{post}$, after that, the branch output $d_v^{post}$ is computed using a feed-forward network (FFN) with layer normalization (LN) and an additional residual connection,as shown in Eqs. (14) to (16):

$$d_i^{post} = \pi \left( \frac{q_v k_t^T}{\sqrt{d/h}} \right) v_t \in \mathbb{R}^{(d/h)}, \tag{14}$$

$$d^{post} = concat(d_1^{post}, d_2^{post}, \ldots, d_h^{post}) \in \mathbb{R}^d, \tag{15}$$

$$d_p^{post} = v_x + FFN\left(LN(v_x + d^{post} W^O)\right) \in \mathbb{R}^d, \tag{16}$$

where $W^O \in \mathbb{R}^{d \times d}$ represents the weight matrix.

**Decoding at the attention level.** We employ attention weights from the three branches to decode composition concepts, as expressed in Eq. (17):

$$Corr(x, (s, o)) = \omega(v_x, p_y^{pre}) + \alpha \cdot \omega(v_x, p_s^{pre}) + \beta \cdot \omega(v_x, p_o^{pre}), \tag{17}$$

where the composition label $y = (s, o)$ and $\omega(v_x, p_y^{pre})$ denotes the weight from the composition branch, $\omega(v_x, p_s^{pre})$ denotes the weight from the state branch, $\omega(v_x, p_o^{pre})$ denotes the weight from the object branch, and $\alpha$ and $\beta$ are hyper-parameters utilized for balancing the loss terms.

Moreover, we devise a cross-entropy loss by assessing the representation quality of the outputs from the three branches, as delineated in Eqs. (18) and (19):

$$a_p = v_x^T d_p^{post} \in \mathbb{R}, \quad \hat{g^E}_i = \frac{\exp(a_i/\tau)}{\sum_j^E \exp(a_j/\tau)} \in \mathbb{R}, \tag{18}$$

$$\begin{aligned} \mathcal{L}_{\mathcal{CA}} = &- \sum_i^{|\mathcal{S}|} \mathbb{I}(i \in s) \log \hat{g^{\mathcal{S}}}_i - \sum_j^{|\mathcal{O}|} \mathbb{I}(j \in o) \log \hat{g^{\mathcal{O}}}_j \\ &- \sum_k^{|\mathcal{Y}|} \mathbb{I}(k \in y) \log \hat{g^{\mathcal{Y}}}_k. \end{aligned} \tag{19}$$

### 3.4 Training and Inference

Finally, we compute the overall model loss using cross-entropy loss as shown in Eq. (20):

$$\mathcal{L} = \mathcal{L}_{\mathcal{CA}} + \gamma \mathcal{L}_{\mathcal{MR}}, \tag{20}$$

where $\gamma$ serves as a balancing parameter to ensure the scales of the individual losses are roughly equivalent.

During inference, given an image $x$, we feed it into the cross-attention branches along with the multiple representations of state primitives, object primitives, and composition concepts. The final inference result is determined by maximizing the correlation score $Corr(x, (s, o))$ by Eq. (21):

$$\begin{aligned} \hat{y} &= \arg \max_{y=(s,o)\in\mathcal{Y}} Corr(x, (s, o)) \\ &= \arg \max_{y=(s,o)\in\mathcal{Y}} \left[\omega(v_x, p_y^{pre}) + \alpha \cdot \omega(v_x, p_s^{pre}) + \beta \cdot \omega(v_x, p_o^{pre})\right], \end{aligned} \tag{21}$$

where $\alpha$ and $\beta$ are hyperparameters for balancing contributions, adaptable across diverse downstream tasks through parameter tuning on different datasets.

## 4 Experiments

### 4.1 Experiments Setting

**Dataset.** We conducted experiments on three datasets: MIT-States [10], UT-Zappos [33], and Clothing16K [34], to assess the model's performance. MIT-States contains general compositional concepts, while Clothing16K focuses specifically on cloth-related compositions, and UT-Zappos focuses on shoe-related compositions. These datasets represent both general and specific domain testing scenarios, with a training-test set split consistent with [4, 8, 23] (see Tab. 1).
**Evaluation metrics.** Zero-shot learning models tend to favor predicting seen classes over unseen classes [2, 20]. To comprehensively assess the model's performance, we employ the AUC metric proposed in [26]. We introduce a bias ranging from $-\infty$ to $+\infty$ for unseen compositions to regulate the model's preference for seen classes. The evaluation metrics we use are as follows: (1) *Best seen accuracy* (S). (2) *Best unseen accuracy* (U). (3) *Best harmonic mean* (H): This metric is the harmonic mean of S and U. (4) *Area under the curve* (AUC): It measures the area under the accuracy curve.

**Table 1: Dataset statistics.**

| | Primitives | | Train | | Val | | | Test | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | state | object | seen | image | seen | unseen | image | seen | unseen | image |
| MIT-States [10] | 115 | 245 | 1262 | 30k | 300 | 300 | 10k | 400 | 400 | 13k |
| UT-Zappos [33] | 16 | 12 | 83 | 23k | 15 | 15 | 3k | 18 | 18 | 3k |
| Clothing16K [34] | 9 | 8 | 18 | 7k | 10 | 10 | 5k | 9 | 8 | 3k |

**Table 2: Closed-world results.** Please refer to Sec. 4.1 for the definition of each evaluation metric, where AUC serves as the primary metric.

| Setting | Method | | MIT-States | | | | UT-Zappos | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | S | U | HM | AUC | S | U | HM | AUC |
| Closed-World | CLIP [27] | *ICML'21* | 30.2 | 46.0 | 26.1 | 11.0 | 15.8 | 49.1 | 15.6 | 5.0 |
| | CoOp [35] | *IJCV'22* | 34.4 | 47.6 | 29.8 | 13.5 | 52.1 | 49.3 | 34.6 | 18.8 |
| | Co-CGE [21] | *TPAMI'22* | 46.7 | 45.9 | 33.1 | 17.0 | 63.4 | 71.3 | 49.7 | 36.3 |
| | ProDA [19] | *CVPR'22* | 37.4 | 51.7 | 32.7 | 16.1 | 63.7 | 60.7 | 47.6 | 32.7 |
| | PromptCVL [32] | *arXiv'22* | 48.5 | 47.2 | 35.3 | 18.3 | 64.4 | 64.0 | 46.1 | 32.2 |
| | CSP [25] | *ICLR'23* | 46.6 | 49.9 | 36.3 | 19.4 | 64.2 | 66.2 | 46.6 | 33.0 |
| | DFSP(i2t) [17] | *CVPR'23* | 47.4 | 52.4 | 37.2 | 20.7 | 64.2 | 66.4 | 45.1 | 32.1 |
| | DFSP(BiF) [17] | *CVPR'23* | 47.1 | 52.8 | 37.7 | 20.8 | 63.3 | 69.2 | 47.1 | 33.5 |
| | DFSP(t2i) [17] | *CVPR'23* | 46.9 | 52.0 | 37.3 | 20.6 | 66.7 | 71.7 | 47.2 | 36.0 |
| | Troika [9] | *CVPR'24* | 49.0 | 53.0 | 39.3 | 22.1 | 66.8 | 73.8 | 54.6 | 41.7 |
| | PLID [1] | *arXiv'23* | 49.7 | 52.4 | 39.0 | 22.1 | 67.3 | 68.8 | 52.4 | 38.7 |
| | GIPCOL [31] | *WACV'24* | 48.5 | 49.6 | 36.6 | 19.9 | 65.0 | 68.5 | 48.8 | 36.2 |
| | PLO-VLM [14] | *arXiv'23* | 49.6 | 52.7 | 39.0 | 22.2 | 67.8 | 75.6 | 53.1 | 42.0 |
| | PLO-LLM [14] | *arXiv'23* | 49.6 | **53.2** | 39.0 | 21.9 | 68.3 | 73.0 | 54.8 | 41.6 |
| | **MRSP(Ours)** | | **53.5** | 52.7 | **41.3** | **23.8** | **71.9** | **81.7** | **65.2** | **52.5** |

**Baselines.** We compare MRSP with some CLIP-based methods on MIT-States and UT-Zappos. The methods including CLIP [27], CoOp [35], Co-CGE [21], ProDA [19], PromptCVL [32], CSP [25], DFSP [17], Troika [9], PLID [1], GIPCOL [31], PLO [14]. The results are shown in Tabs. 2 and 3. We also compare MRSP with some methods on Clothing16K. The methods including SymNet [16], CompCos [20], CGE [23], Co-CGE [21], SCEN [15], IVR [34], OADis [28], ADE [8]. The results are shown in Tab. 4.

**Implementation details.** We conducted experiments on MRSP under two settings: `Closed-world` [22] and `Open-world` [20]. We use the Adam optimizer [12] with a learning rate set to $5 \times 10^{-4}$ and weight decay set to $5 \times 10^{-5}$. During training, the learning rate decays by a factor of 0.8 every 10 epochs. For MIT-States, we set $\alpha = 0.7$, $\beta = 1.3$, $\gamma = 1$, for UT-Zappos, we set $\alpha = 0.7$, $\beta = 0.8$, $\gamma = 1$, and for Clothing16K, we set $\alpha = 1$, $\beta = 1$, $\gamma = 1$. We employ a pre-trained CLIP [7] image encoder concatenated with a 2-layer MLP with a embedding dimension of 1024. The GCN has 2 layers with input dimension 1024, the first layer has a dimension of 8192, and the second layer has a dimension of 1024. Both layers use a dropout probability of 0.5 during training. The number of attention heads in cross-attention is set to 16, and the input and output dimensions are 1024. We train the model with a batch size of 64 and 200 epochs.

### 4.2 Comparison with the SoTA

**Closed-world evaluation.** In Tabs. 2 and 4, we present a comparison between MRSP and state-of-the-art methods in closed-world settings. While MRSP shows a slight 0.5% decrease in best unseen accuracy on the MIT-States dataset, it

**Table 3: Open-world results.** In contrast to Tab. 2, here the prediction space comprises the Cartesian product of the sets of state and object primitives.

| Setting | Method | MIT-States | | | | UT-Zappos | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | S | U | HM | AUC | S | U | HM | AUC |
| Open-World | CLIP [27] *ICML'21* | 30.1 | 14.3 | 12.8 | 3.0 | 15.7 | 20.6 | 11.2 | 2.2 |
| | CoOp [35] *IJCV'22* | 34.6 | 9.3 | 12.3 | 2.8 | 52.1 | 31.5 | 28.9 | 13.2 |
| | Co-CGE [21] *TPAMI'22* | 38.1 | 20.0 | 17.7 | 5.6 | 59.9 | 56.2 | 45.3 | 28.4 |
| | ProDA [19] *CVPR'22* | 37.5 | 18.3 | 17.3 | 5.1 | 63.9 | 34.6 | 34.3 | 18.4 |
| | PromptCVL [32] *arXiv'22* | 48,5 | 16.0 | 17,7 | 6.1 | 64.6 | 44.0 | 37.1 | 21.6 |
| | CSP [25] *ICLR'23* | 46.3 | 15.7 | 17.4 | 5.7 | 64.1 | 44.1 | 38.9 | 22.7 |
| | DFSP(i2t) [17] *CVPR'23* | 47.2 | 18.2 | 19.1 | 6.7 | 64.3 | 53.8 | 41.2 | 26.4 |
| | DFSP(BiF) [17] *CVPR'23* | 47.1 | 18.1 | 19.2 | 6.7 | 63.5 | 57.2 | 42.7 | 27.6 |
| | DFSP(t2i) [17] *CVPR'23* | 47.5 | 18.5 | 19.3 | 6.8 | 66.8 | 60.0 | 44.0 | 30.3 |
| | Troika [9] *CVPR'24* | 48.8 | 18.7 | 20.1 | 7.2 | 66.4 | 61.2 | 47.8 | 33.0 |
| | PLID [1] *arXiv'23* | 49.1 | 18.7 | 20.0 | 7.3 | 67.6 | 55.5 | 46.6 | 30.8 |
| | GIPCOL [31] *WACV'24* | 48.5 | 16.0 | 17.9 | 6.3 | 65.0 | 45.0 | 40.1 | 23.5 |
| | PLO-VLM [14] *arXiv'23* | **49.5** | 18.7 | 20.5 | 7.4 | 68.0 | 63.5 | 47.8 | 33.1 |
| | **MRSP(Ours)** | 46.5 | **23.0** | **22.4** | **8.4** | **69.0** | **65.1** | **52.4** | **38.8** |

**Table 4:** Results on Clothing16K.

| Dataset | Method | Closed-World | | | | Open-world | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | S | U | HM | AUC | S | U | HM | AUC |
| Clothing16K | SymNet [16] *CVPR'20* | 98.0 | 85.1 | 79.3 | 78.8 | 98.2 | 60.7 | 68.3 | 57.4 |
| | CompCos [20] *CVPR'21* | 98.5 | 96.8 | 87.2 | 90.3 | 98.2 | 69.8 | 70.8 | 64.1 |
| | CGE [23] *CVPR'21* | 98.0 | 97.4 | 84.2 | 89.2 | 98.5 | 69.7 | 68.3 | 62.0 |
| | Co-CGE [21] *TPAMI'22* | 98.5 | 94.7 | 87.9 | 88.3 | 98.7 | 63.8 | 69.2 | 59.3 |
| | SCEN [15] *CVPR'22* | 98.0 | 89.6 | 78.5 | 78.8 | 96.7 | 62.3 | 61.5 | 53.7 |
| | IVR [34] *ECCV'22* | 99.0 | 97.0 | 86.6 | 90.6 | 98.7 | 69.0 | 72.0 | 63.6 |
| | OADis [28] *CVPR'23* | 97.7 | 94.2 | 86.1 | 88.4 | 98.0 | 58.6 | 63.2 | 53.4 |
| | ADE [8] *CVPR'23* | 98.2 | 97.7 | 88.7 | 92.4 | 99.0 | 73.1 | 74.2 | 68.0 |
| | **MRSP(Ours)** | **99.0** | **99.7** | **95.5** | **96.4** | **99.7** | **88.4** | **89.1** | **86.0** |

exhibits overall performance improvements across all other metrics. For example, compared to PLO-LLM, MRSP achieves a 7.8% increase in best unseen accuracy, a 5.9% increase in HM, and an 8.7% increase in AUC. Particularly noteworthy is MRSP's significant outperformance of the previous state-of-the-art PLO-VLM method on the UT-Zappos dataset, with a remarkable 25% improvement in AUC performance and a first-time elevation of best unseen accuracy to 81.7%. This substantial enhancement indicates that MRSP greatly improves conceptual understanding accuracy. Similar occurrences are observed on the Clothing16K dataset, MRSP emerges with all metrics soaring above the 95% mark for the very first time. Remarkably, MRSP attains stellar scores of 99.0% for best seen accuracy and an impressive 99.7% for best unseen accuracy, underscoring the significant prowess of MRSP in tackling compositional discrimination tasks.

**Table 5:** Experiments with different backbones on the MIT-States dataset.

| Backbone | Methods | MIT-States | | | |
|---|---|---|---|---|---|
| | | S | U | HM | AUC |
| ResNet18 | CGE [23] | 32.8 | 28.0 | 21.4 | 8.6 |
| | MRSP | 38.3 | 36.8 | 24.8 | 9.9 |
| ViT-B/32 | DFSP [17] | 36.7 | 43.4 | 29.4 | 13.2 |
| | PLO-VLM [14] | 41.1 | 44.2 | 31.3 | 14.8 |
| | PLO-LLM [14] | 44.2 | 47.9 | 34.3 | 17.4 |
| | MRSP | 49.8 | 48.9 | 37.4 | 20.0 |
| ViT-L/14 | DFSP [17] | 46.9 | 52.0 | 37.3 | 20.6 |
| | PLO-VLM [14] | 49.6 | 52.7 | 39.0 | 22.2 |
| | PLO-LLM [14] | 49.6 | **53.2** | 39.0 | 21.9 |
| | MRSP | **53.5** | 52.7 | **41.3** | **23.8** |

**Table 6:** Ablation study on MIT-States. a) MR. ✓: Utilizing $\{s\ s^y\ y\ o^y\ o\}$ nodes for representation, ✗: Using only $\{s\ y\ o\}$ nodes; b) cross-attention (CA). ✓: Employing cross-attention for decoding, ✗: Using 3-layer MLP decoding.

| Component | | MIT-States | | | |
|---|---|---|---|---|---|
| MR | CA | S | U | HM | AUC |
| ✗ | ✗ | 34.8 | 30.9 | 23.1 | 8.2 |
| ✓ | ✗ | 49.1 | 51.1 | 38.5 | 20.0 |
| ✗ | ✓ | 42.5 | 40.0 | 28.6 | 12.6 |
| ✓ | ✓ | **53.5** | **52.7** | **41.3** | **23.8** |

**Open-world evaluation.** In Tabs. 3 and 4, we provide a comparison between MRSP and other methods in the open-world setting. Despite a 3% decrease in best seen accuracy on the MIT-States dataset, our approach surpasses various previous methods across all metrics. For example, compared to the state-of-the-art PLO-VLM, MRSP achieves a 7.2% increase in AUC on MIT-States and an impressive 25% increase on UT-Zappos. In comparison to ADE, our method enhances the AUC metric by 26.5% on Clothing16K. Furthermore, when examining the performance degradation of different methods transitioning from the closed-world to open-world settings, we are pleased to observe that on the Clothing16K dataset, MRSP experiences only a 10.7% decrease in AUC, whereas previous methods suffer a decrease of at least 26.4% (ADE) and up to 39.6% (OADis). This underscores the robustness of MRSP in specific domain tasks.

### 4.3 Ablation Studies

**Backbone: ResNet and ViT.** The quality of node feature vectors significantly impacts MRSP model performance, as evidenced by our comparisons using different backbones (see Tab. 5). As expected, models utilizing ViT outperformed those using ResNet18, demonstrating improvements of 102% (ViT-B/32) and 240% (ViT-L/14), respectively. Furthermore, despite not employing a more efficient encoder, MRSP still achieved a 15.1% improvement in performance (measured by AUC) compared to the CGE method with the same configuration. This underscores MRSP's robustness to lower-quality encoding results.

**Module effects.** We evaluated the effectiveness of multi-representation (MR) and cross-attention decoders, with results presented in Tab. 6. The adoption of the MR strategy yielded an 11.2% improvement in AUC performance compared to using a single representation, underscoring the enhanced conceptual understanding achieved by MR when combined with cross-attention. Moreover, even with a single representation, employing cross-attention instead of a three-layer MLP decoder resulted in a 4.4% increase in AUC performance, highlighting the efficiency of attention-based decoders in concept recognition tasks.

**Table 7:** Experiments with different setting of GCN.

| GCN Settings | | MIT-States | | | |
|---|---|---|---|---|---|
| | | S | U | HM | AUC |
| Layers | $d4096, d1024$ | 50.6 | 48.8 | 37.8 | 20.9 |
| | $d4096 \times 2, d1024$ | 50.4 | 51.3 | 38.8 | 21.3 |
| | $d4096 \times 3, d1024$ | 47.9 | 52.1 | 37.4 | 20.1 |
| | $d8192, d1024$ | **53.5** | 52.7 | 41.3 | **23.8** |
| | $d8192 \times 2, d1024$ | 50.8 | 53.5 | 39.8 | 22.5 |
| LN | no norm | 48.7 | 52.5 | 39.4 | 20.0 |
| | post-norm | 53.1 | **54.5** | **42.2** | 22.5 |
| | pre-norm | **53.5** | 52.7 | 41.3 | **23.8** |

**Table 8:** Experiments with different setting of cross-attention (CA).

| CA Settings | | MIT-States | | | |
|---|---|---|---|---|---|
| | | S | U | HM | AUC |
| MLP | 1 *layer* | 52.6 | 51.7 | 40.8 | 23.4 |
| | 2 *layers* | **53.5** | **52.7** | **41.3** | **23.8** |
| | 3 *layers* | 50.9 | 50.0 | 38.9 | 22.5 |
| Inference | $y + \delta \cdot s \cdot o$ | 47.4 | 52.1 | 38.9 | 19.7 |
| | $y + \alpha \cdot s + \beta \cdot o$ | **53.5** | **52.7** | **41.3** | **23.8** |
| Heads | 8 *heads* | 52.3 | 50.9 | 39.9 | 22.4 |
| | 16 *heads* | **53.5** | **52.7** | **41.3** | **23.8** |
| | 32 *heads* | 52.2 | 51.1 | 39.5 | 22.0 |


**(a)** image-to-text retrieval **(b)** text-to-image retrieval

**Fig. 5: Retrieval.** We denote state primitives in orange font, object primitives in blue font, correct classifications in green font (box), and incorrect classifications in red font (box), (a) shows retrieval on MIT-States and (b) shows retrieval on Clothing16K.

**Adjustments on GCN.** We adjusted several key parameters of GCN and summarized the results in Tab. 7. In this table, 'd' denotes the use of dropout, while the accompanying numbers represent the dimensions of node features for each layer. Generally, larger feature dimensions tend to enhance performance, as evidenced by a notable 13.8% increase under the 2-layer GCN setting. However, performance varies across different GCN depths due to uncertainties in determining the neighborhood scope of nodes in current compositional graph generation strategies. For instance, under the 8192-dimensional setting, the *best unseen accuracy* of a 3-layer GCN reaches a maximum of 53.5, whereas the AUC indicator of a 2-layer GCN peaks at 22.5. Future research may explore more controllable and effective methods for compositional graph generation to enhance graph network performance. Lastly, the utilization of pre-norm [29] resulted in a significant 19% improvement in GCN performance.

**Adjustments on cross-attention.** We conducted fine-tuning on several key parameters of cross-attention and summarized the results in Tab. 8. The model's performance showed slight fluctuations with changes in the number of MLP layers and attention heads (up to 5.7% in the MLP setting and 8.1% in the attention head setting). Additionally, we explored two different inference approaches. Initially, we utilized the formula $\omega(v_x, p_y^{pre}) + \delta\omega(v_x, p_s^{pre}) \cdot \omega(v_x, p_o^{pre})$ for calculating

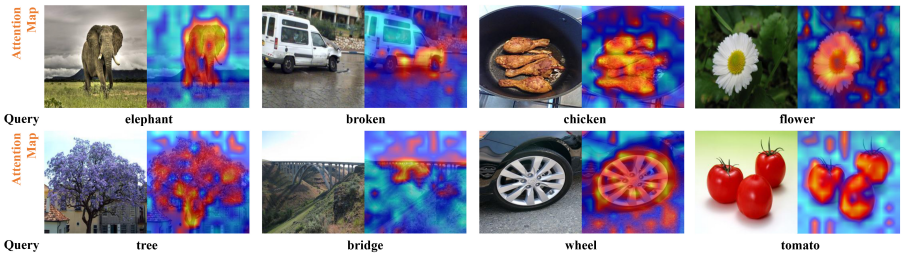



**Fig. 6: Attention Visualization.** Images were randomly selected from MIT-States. Below images is the query input, and on the right side is the attention map.

$Corr(x, (s, o))$, with $\delta = 5$. However, due to the significant scale change induced by multiplication, it proved challenging to control.

### 4.4 Visualization

**Retrieval experiments.** We randomly sampled 9 images in MIT-States and sorted the $Corr$ scores with different compositions, as shown in Fig. 5a. Ensuring that the *top-1* result is correct presents a challenge. However, the model could accurately identify the *top-3* compositions in most cases. In the Clothing16K, we selected several labels and sorted their $Corr$ scores with different images, as shown in Fig. 5b. Only occasional failures were observed, particularly in cases like "*green suit*", affected by environmental lighting and closely resembling categories such as "*shorts*" and "*skirt*".

**Attention Visualization.** We randomly selected 8 images and used their corresponding labels as queries to retrieve model responses, as depicted in Fig. 6. The model's regions of interest are highlighted in red, while blue areas indicate regions the model considers unrelated to the given query. This demonstrates MRSP's ability to accurately comprehend concepts.

## 5 Conclusion

In this work, we address the challenges of CZSL and propose MRSP as a novel solution to mitigate conditional variance by providing multiple representations for individual primitives. Our approach utilizes GCN to construct these multiple representations and employs cross-attention for attention-level decoding. By integrating knowledge of states, object primitives, and compositions into the model's inference process, MRSP significantly enhances concept understanding and composition discrimination accuracy. Experimental results on the MIT-States, UT-Zappos, and Clothing16K datasets demonstrate MRSP's superiority over state-of-the-art methods in both closed-world and open-world settings. However, it's important to note that while MRSP enhances conceptual understanding, it also increases computational and storage demands. Our future work will focus on exploring more efficient approaches to multiple representations construction.

## References


1. Bao, W., Chen, L., Huang, H., Kong, Y.: Prompting language-informed distribution for compositional zero-shot learning. arXiv preprint arXiv:2305.14428 (2023)
2. Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 52–68. Springer (2016)
3. Chen, C.F.R., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 357–366 (2021)
4. Chen, H., Jiang, J., Zheng, N.: Learning to infer unseen single-/multi-attribute-object compositions with graph networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
5. Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks. In: International conference on machine learning. pp. 1725–1735. PMLR (2020)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Fang, A., Jose, A.M., Jain, A., Schmidt, L., Toshev, A., Shankar, V.: Data filtering networks. arXiv preprint arXiv:2309.17425 (2023)
8. Hao, S., Han, K., Wong, K.Y.K.: Learning attention as disentangler for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15315–15324 (2023)
9. Huang, S., Gong, B., Feng, Y., Lv, Y., Wang, D.: Troika: Multi-path cross-modal traction for compositional zero-shot learning. arXiv preprint arXiv:2303.15230 (2023)
10. Isola, P., Lim, J.J., Adelson, E.H.: Discovering states and transformations in image collections. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1383–1391 (2015)
11. Karthik, S., Mancini, M., Akata, Z.: Kg-sp: Knowledge guided simple primitives for open world compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9336–9345 (2022)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
14. Li, L., Chen, G., Xiao, J., Chen, L.: Compositional zero-shot learning via progressive language-based observations. arXiv preprint arXiv:2311.14749 (2023)
15. Li, X., Yang, X., Wei, K., Deng, C., Yang, M.: Siamese contrastive embedding network for compositional zero-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9326–9335 (2022)
16. Li, Y.L., Xu, Y., Mao, X., Lu, C.: Symmetry and group in attribute-object compositions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11316–11325 (2020)
17. Lu, X., Guo, S., Liu, Z., Guo, J.: Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23560–23569 (2023)

18. Lu, X., Liu, Z., Guo, S., Guo, J., Huo, F., Bai, S., Han, T.: Drpt: Disentangled and recurrent prompt tuning for compositional zero-shot learning. arXiv preprint arXiv:2305.01239 (2023)
19. Lu, Y., Liu, J., Zhang, Y., Liu, Y., Tian, X.: Prompt distribution learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5206–5215 (2022)
20. Mancini, M., Naeem, M.F., Xian, Y., Akata, Z.: Open world compositional zero-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5222–5230 (2021)
21. Mancini, M., Naeem, M.F., Xian, Y., Akata, Z.: Learning graph embeddings for open world compositional zero-shot learning. IEEE Transactions on pattern analysis and machine intelligence (2022)
22. Misra, I., Gupta, A., Hebert, M.: From red wine to red tomato: Composition with context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1792–1801 (2017)
23. Naeem, M.F., Xian, Y., Tombari, F., Akata, Z.: Learning graph embeddings for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 953–962 (2021)
24. Nagarajan, T., Grauman, K.: Attributes as operators: factorizing unseen attribute-object compositions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 169–185 (2018)
25. Nayak, N.V., Yu, P., Bach, S.H.: Learning to compose soft prompts for compositional zero-shot learning. arXiv preprint arXiv:2204.03574 (2022)
26. Purushwalkam, S., Nickel, M., Gupta, A., Ranzato, M.: Task-driven modular networks for zero-shot compositional learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3593–3602 (2019)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
28. Saini, N., Pham, K., Shrivastava, A.: Disentangling visual embeddings for attributes and objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13658–13667 (2022)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
30. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International conference on machine learning. pp. 6861–6871. PMLR (2019)
31. Xu, G., Chai, J., Kordjamshidi, P.: Gipcol: Graph-injected soft prompting for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5774–5783 (2024)
32. Xu, G., Kordjamshidi, P., Chai, J.: Prompting large pre-trained vision-language models for compositional concept learning. arXiv preprint arXiv:2211.05077 (2022)
33. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 192–199 (2014)
34. Zhang, T., Liang, K., Du, R., Sun, X., Ma, Z., Guo, J.: Learning invariant visual representations for compositional zero-shot learning. In: European Conference on Computer Vision. pp. 339–355. Springer (2022)

35. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022)